

Oficina de Análise de Requisitos e IHC baseada em estilos de aprendizagem

Roberto Muñoz¹

René Noël¹

Thiago Barcelos²

Virgínia Chalegre³

Natali Rios¹

¹ Universidad de Valparaíso
Escuela de Ingeniería Civil en
Informática
Valparaíso, Chile

{roberto.munoz.s; rene.noel}@uv.cl;
natali.rios@alumnos.uv.cl

² Instituto Federal de Educação,
Ciência e Tecnologia de São
Paulo
São Paulo, Brasil

tsbarcelos@ifsp.edu.br

³ CESAR.EDU
Recife, Brasil
vivichalegre@gmail.com

RESUMO

Este artigo apresenta o desenvolvimento de um curso prático de Interação Humano-Computador, integrado ao estudo de Engenharia de Requisitos, em uma universidade chilena. A disciplina tem como objetivo explorar as relações interdisciplinares entre as duas áreas, mostrando como elas podem ser complementares. Propõe-se a identificação dos estilos de aprendizagem dos alunos como uma estratégia para potencializar a adaptação aos diversos papéis no processo de desenvolvimento de software. São apresentados os objetivos de investigação que estão sendo explorados durante o primeiro oferecimento da disciplina, bem como os resultados esperados.

Palavras-chave

Ensino de IHC, Análise de Requisitos, Oficina de IHC, Integração de Disciplinas, Experiência Prática.

INTRODUÇÃO

O curso de graduação de *Ingeniería Civil en Informática*, oferecido pela Universidade de Valparaíso, tem uma grade curricular composta por disciplinas orientadas à formação de engenheiros especializados em software, tendo em seu escopo diversos tópicos da área de Engenharia de Software [1]. O projeto da grade curricular inclui oito disciplinas relacionadas a esses tópicos, além de uma disciplina prática, no formato de oficina, que busca a integração de todos esses conceitos. Na experiência dos autores, a inter-relação entre disciplinas oferecidas de forma independente não é satisfatória. Isso se aplica a disciplinas como Análise de Requisitos, Projeto e Arquitetura de Software e Projeto de Interface Humano-Computador; tal constatação também é reportada em instituições semelhantes [3]. Para uma melhor integração entre as diversas atividades da Engenharia de Software, a escola está atualmente reformulando algumas de suas disciplinas de modo a fomentar a integração entre elas ao longo do processo de formação dos alunos. Com esse objetivo, novas disciplinas, no formato de oficina, estão sendo desenvolvidas, para que a integração interdisciplinar entre os tópicos possa acontecer em paralelo à formação.

Este trabalho apresenta uma das iniciativas da universidade para antecipar a integração entre conteúdos da Engenharia de Software e complementar a formação em Análise de Requisitos com técnicas do Projeto de Interação Humano-Computador. Uma motivação para essa iniciativa é a similaridade encontrada entre as técnicas ágeis para Análise de Requisitos cobertas pelo curso e técnicas existentes de Design Centrado no Usuário (DCU), similaridade essa já abordada anteriormente por outros autores [10,17,18]. Por outro lado, pretende-se utilizar a oficina como espaço para explorar algumas hipóteses de pesquisa que relacionam os estilos de aprendizagem dos alunos com a qualidade dos artefatos produzidos através de técnicas utilizadas para o DCU.

Assim, o artigo está organizado da seguinte forma: primeiramente, é apresentada a situação atual do ensino de IHC no Chile e na Universidade de Valparaíso; também é descrita a estrutura da disciplina de Análise de Requisitos na universidade. Posteriormente, é detalhada a motivação para o projeto da oficina e sua condução em paralelo à disciplina de Análise. Na sequência, a estrutura da oficina é detalhada e os objetivos de investigação propostos são apresentados. Finalmente, são apresentados os resultados esperados da oficina, juntamente com as conclusões e trabalhos futuros.

SITUAÇÃO ATUAL DO ENSINO DE IHC NO CHILE

As atividades de IHC no Chile se iniciaram em 1994 e até o presente momento não se encontram consolidadas. Sua aplicação nas empresas é baixa e, nas universidades que oferecem disciplinas na área, sua integração ao currículo está sendo feita [6]. Atualmente as principais universidades oferecem em seu currículo disciplinas relacionadas com IHC, porém na maioria dos casos se tratam de disciplinas optativas. Uma problemática existente é que os professores, envolvidos na elaboração dos programas curriculares, acreditam que IHC não é relevante em comparação com outras disciplinas; dessa forma, o currículo enfatiza os aspectos internos de funcionamento dos sistemas, ignorando a importância da interface e, principalmente, do usuário [3].

Situação atual da Universidade de Valparaíso

Como mencionado anteriormente, a graduação em *Ingeniería Civil en Informática* oferece uma sequência de disciplinas na área de Engenharia de Software. Os alunos cursam no quinto semestre a disciplina de Fundamentos de Engenharia de Software; no sexto semestre, a disciplina de Metodologias de Análise; no sétimo semestre, as disciplinas de Metodologia de Projeto e Interação Humano-Computador; entre o nono e o décimo semestres, cursam Gestão de Projetos de Informática e Oficina de Aplicações. Essas duas últimas consistem no planejamento e execução de um projeto de desenvolvimento de software em todas as suas etapas. É permitida ainda a opção por três diferentes ênfases: Gestão e Projeto de Bases de Dados, Redes e Telecomunicação e Gestão de Projetos de Software, que se iniciam a partir do nono semestre do curso.

A disciplina de Interação Humano-Computador é obrigatória na grade curricular. Seu objetivo principal é apresentar aos alunos os conceitos teóricos básicos da área, em especial os relacionados a fatores humanos (atenção, memória, cognição), bem como os princípios de usabilidade. Também pretende-se capacitar os alunos a utilizar metodologias de desenvolvimento adequadas para a construção de software interativo com boa usabilidade. O curso prevê uma carga horária de 16 semanas de aula, com três horas por semana. A carga horária da disciplina tem permitido a apresentação dos conceitos fundamentais da área, porém não é possível oferecer uma experiência prática mais aprofundada aos alunos.

A disciplina de Análise de Requisitos

Os tópicos pertinentes à Análise de Requisitos estão presentes no curso em uma disciplina do sexto semestre, denominada Metodologias de Análise, que tem como objetivo levar o aluno a “compreender o processo de análise dentro do contexto do ciclo de vida do software, conhecer e aplicar os princípios e fundamentos do processo de análise, bem como metodologias de apoio ao processo de análise”. A disciplina foi projetada tendo em mente o desenvolvimento incremental de habilidades técnicas para elicitación, análise e especificação de requisitos através de diversos métodos, que vão desde os ágeis [4,27,13], envolvendo a participação ativa do cliente, com técnicas para apoiar a análise e identificação de novos requisitos, como prototipação e modelagem ágil [2], chegando à elaboração de diagramas e modelos utilizando a UML 2.0, para uma especificação mais rigorosa dos requisitos.

Os conteúdos da disciplina são apresentados em 10 semanas com 3 horas de aulas por semana, distribuídas em 1,5 horas de aulas expositivas e 1,5 horas de trabalho prático em sala. Além disso, 5 semanas adicionais são reservadas para a apresentação dos avanços dos alunos em um projeto com equipes, realizado fora do horário de aulas, em que as técnicas aprendidas devem ser aplicadas. O projeto tem como objetivo produzir uma especificação de requisitos inicial de um software, idealizado pelos próprios alunos,

que será construído em disciplinas futuras.

PERFIL DOS ALUNOS E TAREFAS DE PROJETO DE IHC

É necessário considerar que os alunos recebem e processam informações de diferentes maneiras: vendo ou ouvindo, refletindo ou atuando, raciocinando lógica ou intuitivamente, analisando ou visualizando. Os estudantes, e em geral todas as pessoas, são mais eficazes quando participam de atividades didáticas que se aproximam do seu estilo pessoal de aprendizagem [7]. Felder e Silverman [8] propõem um modelo de estilos de aprendizagem para o ensino de Engenharia que classifica os indivíduos em quatro escalas independentes, a seguir:

- *Ativos* (ATI), que aprendem manipulando coisas e trabalhando em equipe ou *Reflexivos* (REF), que aprendem pensando sobre as coisas e trabalhando individualmente;
- *Sensitivos* (SEN), que são concretos, práticos, orientados ao fazer e aos procedimentos ou *Intuitivos* (INT), que são conceituais, inovadores, orientados às teorias;
- *Visuais* (VIS), que preferem a apresentação visual de material como filmes, tabelas ou diagramas de fluxos ou *Verbais* (VER), que preferem as explicações escritas ou faladas;
- *Sequenciais* (SEQ), que aprendem pouco a pouco, de forma ordenada, ou *Globais* (GLO), que aprendem de forma holística.

A escala de Myers-Briggs [19] pode ser utilizada em uma análise mais ampla, que considera não somente as atividades e processos cognitivos, relacionados à aprendizagem, mas a personalidade do indivíduo como um todo. A escala caracteriza a personalidade através de quatro dicotomias: Extroversão (E) ou Introversão (I); Sensorial (S) ou Intuitivo (N); Razão (T – *thinking*) ou Emoção (F – *feeling*); Julgamento (J) ou Percepção (P).

Com o objetivo de fornecer subsídios iniciais para a definição da nova oficina, foi aplicado o questionário de avaliação dos estilos de aprendizagem proposto por [9] a 15 alunos da disciplina de Metodologia de Análise. Os resultados da pesquisa são apresentados na Tabela 1. Os mesmos alunos responderam o instrumento proposto por Myers-Briggs, cujos resultados são apresentados na Tabela 2.

Os resultados indicam que pode haver um perfil heterogêneo de estilos de aprendizagem entre os alunos, considerando uma pesquisa mais abrangente no futuro. Há uma pequena predominância por alunos reflexivos (62%), sensitivos (92%), visuais (77%) e globais (62%). Na avaliação de estilos de aprendizagem, a maior parte dos alunos foram classificados no perfil ISTJ, que corresponde às características: Introspectivo, Sensitivo, Racional, Julgador. Segundo McConnell [16], esse perfil de

personalidade predomina entre os desenvolvedores de software, sendo aplicável a até 40% deles.

Tabela 1. Classificação de estilos de aprendizagem

	<i>ATI</i>	<i>REF</i>	<i>SEN</i>	<i>INT</i>	<i>VIS</i>	<i>VER</i>	<i>SEQ</i>	<i>GLO</i>
Porc.	38%	62%	92%	8%	77%	23%	38%	62%

Tabela 2. Classificação de estilos de personalidade

ISTJ	ISFJ	INFJ	INTJ
26,7%	13,3%	6,7%	0,0%
ISTP	ISFP	INFP	INTP
6,7%	6,7%	6,7%	6,7%
ESTP	ESFP	ENFP	ENTP
0,0%	6,7%	0,0%	0,0%
ESTJ	ESFJ	ENFJ	ENTJ
6,7%	0,0%	6,7%	6,7%

Uma breve análise dos estilos de aprendizagem e dos perfis de personalidade dos alunos permite conjecturar algumas implicações ao desenvolvimento de uma oficina de IHC, considerando as principais atividades do processo de *design* de interação. Por exemplo, os estudantes que têm preferência pela aprendizagem sequencial e visual poderiam realizar uma observação em campo de forma mais adequada e com resultados mais completos; alunos com perfil fortemente global poderiam produzir melhores protótipos do tipo *storyboard*, devido ao fato de terem uma visão geral da interação; alunos com perfil verbal teriam um melhor aproveitamento ao conduzir entrevistas com usuários, e assim por diante.

Entretanto, apesar da predominância de alguns perfis nos alunos, os demais perfis não devem ser ignorados; dessa forma, o projeto da oficina envolverá o desenvolvimento de atividades e materiais adequados a cada estilo de aprendizagem.

INTEGRANDO ANÁLISE DE REQUISITOS E IHC: MOTIVAÇÃO

As principais disciplinas de grade curricular de *Ingeniería en Informática* que tem seu foco em tópicos da Engenharia de Software mencionam que existem formas de aplicar conceitos de IHC ao desenvolvimento de software, mas não é ensinado como aplicá-los a um projeto real. Isso se reflete nos resultados das atividades práticas dos alunos nas disciplinas integradoras, onde os conteúdos e técnicas de IHC não são aplicados. Essa constatação também foi relatada em outros contextos, tais como [3]. Apesar da existência de uma disciplina de IHC na grade, ela aborda apenas os conceitos fundamentais da área.

Apesar da disciplina Metodologias de Análise não ter em seu escopo o projeto e construção de software, as técnicas ágeis para análise de requisitos apresentadas tem características similares às técnicas de DCU. Ambas têm natureza iterativa, são centradas na perspectiva do usuário sobre o sistema, e agregam valor, através do *feedback* contínuo, seja sobre o próprio software, no caso do desenvolvimento ágil, ou sobre protótipos de baixa

fidelidade, no caso do DCU [11]. A utilização de técnicas de DCU que integram o levantamento de requisitos, contexto de utilização e perfil do usuário pode apoiar o ensino da Engenharia de Requisitos, conforme discutido por Barcelos e Matos [3]. Uma revisão bibliográfica de trabalhos envolvendo a integração de DCU e desenvolvimento ágil [10,17,18,21] motivou então a criação de uma oficina de desenvolvimento de sistemas envolvendo o projeto de IHC com as seguintes características:

- O material de apoio será produzido de acordo com os estilos de aprendizagem predominantes nos alunos participantes com o objetivo de facilitar o processo de ensino-aprendizagem [7];
- As funções assumidas pelos alunos nas atividades da disciplina de Metodologias de Análise e da oficina serão, em primeira instância, definidas de acordo com seus estilos de aprendizagem e personalidades;
- A oficina envolverá as fases próprias do processo de desenvolvimento de interfaces centrado no usuário. As atividades de Análise e Projeto serão inseridas nessas fases de forma coordenada com as técnicas ágeis de elicitação de requisitos abordadas na disciplina Metodologia de Análise;
- Os alunos deverão desenvolver um projeto com clientes reais ao longo da disciplina de análise e da oficina. O projeto terá cinco entregas sincronizadas com a disciplina e a oficina;
- Ao final da oficina, os alunos deverão ter um protótipo final da interface com o usuário, enquanto que ao final da disciplina de análise deverão ter os requisitos funcionais e não funcionais analisados e especificados;
- Tanto o protótipo quanto a especificação de requisitos servirão de base para uma próxima disciplina, em que se trabalhará a modelagem e construção do software;
- A oficina terá o caráter de disciplina optativa, ou seja, nem todos os alunos que se matriculam na disciplina de análise serão obrigados a se matricular na oficina.

PROJETO DA OFICINA DE IHC

A partir da revisão bibliográfica efetuada, da análise da grade curricular atual e do levantamento dos estilos de aprendizagem, projetou-se uma oficina de IHC com a colaboração das experiências de colegas de instituições de ensino brasileiras. A oficina, de caráter teórico e prático, tem como objetivo aprofundar os principais conceitos de IHC e sua aplicação no desenvolvimento de software.

A nova oficina de IHC proposta será oferecida como uma disciplina eletiva de ênfase no nono semestre pois, nesse momento, o aluno idealmente já terá cursado a formação em Engenharia de Software e vivenciado o desenvolvimento de um projeto em todas as suas etapas (análise, projeto, implementação, testes e implantação).

Adicionalmente, os alunos que cursam Metodologias de Análise no sexto semestre poderão também cursar a disciplina, visto que se espera sensibilizá-los para a importância de IHC em etapas prévias de sua formação. O motivo é evidenciar a importância do usuário (e, conseqüentemente da disciplina de IHC) no processo de análise de requisitos.

Pretende-se que os alunos possam conhecer métodos de IHC, compreender os seus princípios e aplicá-los ao desenvolvimento de sistemas. Ainda, pretende-se que os alunos possam identificar adequadamente quais métodos se aplicam a cada etapa do desenvolvimento de software. Para o projeto da oficina, por limitações de tempo, foi selecionado um subconjunto de técnicas: observação participativa [26], *card sorting* [25], produção de *storyboards* e protótipos em papel [24], protótipos em vídeo [15], avaliação heurística [20] e avaliação baseada em pensamento em voz alta [12], procurando expor sua contextualização às etapas do processo de desenvolvimento de software.

Características da oficina

A oficina abrirá inscrições para no máximo 20 alunos, com o requisito mínimo de já terem cursado as disciplinas de Fundamentos de Engenharia de Software e Interface Homem-Máquina. Os alunos se dividirão em grupos de 4 pessoas, onde cada um deles terá um papel a desempenhar. Então, os papéis serão alternados entre os alunos, à medida que avançam as atividades da oficina. Com isso pretende-se que os alunos possam assumir adequadamente diferentes papéis em uma equipe de desenvolvimento.

Essa estratégia se baseia na teoria do construtivismo social, na qual o conhecimento é uma construção elaborada por cada indivíduo através de sua interação com o entorno e as relações sociais. De acordo com [14], todos os processos psicológicos superiores (comunicação, linguagem, raciocínio, etc.) se adquirem primeiramente em um contexto social para depois serem internalizados. Essa internalização é um produto do uso de um determinado comportamento cognitivo em um contexto social. Como os alunos em grupo terão que desempenhar diferentes papéis ao longo da disciplina, poderão aprender a partir das atividades dos companheiros, dado que em um primeiro momento cada aluno irá desempenhar o papel ao qual melhor se adequa, de acordo com a sua preferência. Segundo [5,22] o desenvolvimento da inteligência está associado a diferentes níveis de estruturas mentais; assim, com o revezamento de papéis entre os alunos espera-se que eles possam adquirir novos esquemas e estruturas mentais.

Estrutura didática da oficina

A oficina alternará aulas teóricas e práticas e terá como objetivo final o desenvolvimento do protótipo de uma interface humano-computador através das etapas do processo de desenvolvimento de software (análise, design, implementação, avaliação e implantação), incorporando a

estas as técnicas de DCU [11]. Os alunos deverão desenvolver habilidades para interagir com diferentes *stakeholders* e identificar as necessidades do usuário final a partir do enfoque das áreas multidisciplinares de IHC (psicologia, ergonomia, cognição, entre outras). Assim, pretende-se que possam desenvolver um enfoque distinto do enfoque puramente lógico que estão habituados a utilizar em atividades de engenharia [23]. Além disso, pretende-se desenvolver a capacidade criativa dos alunos para identificar as demandas para a criação ou melhoria de sistemas computacionais.

A oficina foi desenvolvida em torno de cinco unidades principais, representando as etapas do processo de desenvolvimento de software, e mais uma unidade conceitual inicial. As atividades estão distribuídas em uma carga horária total de 16 aulas, com três horas cada. Cada aula está associada a uma das etapas, conforme a descrição a seguir:

Aula 1: Introdução. Na primeira aula se realizará uma introdução ao curso, com a apresentação dos objetivos do mesmo. Será feita uma revisão dos conceitos fundamentais de IHC, a importância da disciplina, definições de DCU, usabilidade e acessibilidade.

Aula 2: Etapa de Análise. Aula expositiva de introdução à etapa, explanação sobre os métodos a aplicar.

Aulas 3 e 4: Etapa de Análise. Os alunos deverão definir o projeto que irão desenvolver. Deverão definir o grupo de trabalho e começar o planejamento e aplicação dos métodos de entrevistas, observação de campo, e identificação de *personas*. Na aula seguinte são aplicados os métodos de tempestade de ideias (*brainstorm*), *card sorting* e identificação de casos de uso essenciais.

Aulas 5 e 6: Etapas de Projeto e Avaliação. Serão realizadas duas aulas expositivas com uma introdução aos métodos aplicáveis a cada uma das etapas, em preparação às iterações que irão acontecer nas aulas seguintes.

Aula 7: Etapa de Projeto. Os alunos trabalharão no desenvolvimento de *storyboards* e protótipos de papel.

Aula 8: Etapa de Avaliação. Nesta aula serão aplicados métodos de avaliação aos protótipos de papel desenvolvidos na aula anterior.

Aula 9: Etapa de Projeto. Os alunos desenvolverão um protótipo digital, refinado a partir dos resultados da avaliação feita na aula anterior.

Aula 10: Etapa de Avaliação. Os alunos aplicarão métodos de avaliação ao protótipo digital.

Aula 11: Etapa de Implementação. Os alunos implementam um “esqueleto” da interface em uma linguagem de programação escolhida, mas ainda sem funcionalidade.

Aula 12: Etapa de Avaliação. Os alunos aplicam métodos de avaliação ao “esqueleto” da interface.

Aula 13 e 14: Etapa de Implementação. Os alunos desenvolvem a implementação das funcionalidades na linguagem de sua escolha.

Aula 15: Etapa de Avaliação. Os alunos aplicam métodos de avaliação à implementação final da interface.

Aula 16: Etapa de Entrega. Será realizada uma aula expositiva sobre os métodos que poderão ser aplicados a essa etapa.

A Figura 1 mostra o planejamento das aulas e a sua característica iterativa, de acordo com a metodologia de Design Centrado no Usuário.

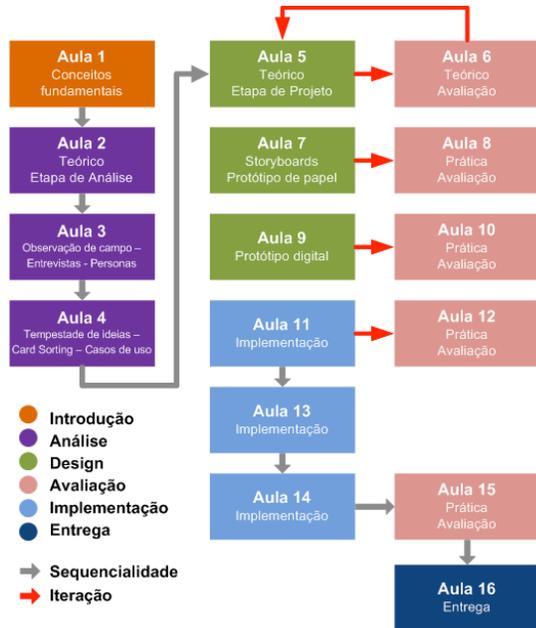


Figura 1. Estrutura das aulas

A avaliação será realizada mediante a entrega de quatro artefatos de projeto. As entregas serão paralelas às entregas solicitadas na disciplina de Metodologias de Análise. O conteúdo dos artefatos a serem desenvolvidos na oficina e na disciplina de requisitos é detalhado na Tabela 3.

Objetivos de investigação

A oficina está sendo oferecida aos alunos pela primeira vez no segundo semestre de 2012. A partir dos artefatos produzidos pelos alunos e de suas impressões acerca da oficina pretende-se obter dados e observações para explorar as seguintes hipóteses de pesquisa:

- O desenvolvimento paralelo de requisitos de software e do projeto de usabilidade leva a um produto de melhor qualidade em ambas as dimensões;
- Há diferenças qualitativas no resultado da aplicação de técnicas de DCU em um projeto de software, dependendo do estilo de aprendizagem e da personalidade de quem conduz as aulas.

Tabela 3. Artefatos produzidos pelos alunos

ID	Conteúdo - Oficina IPC	Conteúdo - Disciplina de Requisitos
1	Resultados de entrevistas, <i>card sorting</i> , Observação de Campo (Etnográfica), Definição de <i>personas</i> , Tempestade de Ideias.	Modelo de Domínio com CRC, <i>Personas</i> , Histórias de Usuário.
2	Cenários, Protótipo Inicial, Vídeo Protótipo, Árvores de Menu, Protótipo de Alto Nível.	Protótipo, Requisitos Orientados a Metas.
3	Resultados de teste de usabilidade (pensando em voz alta), Avaliação Heurística, Rota Cognitiva, Experimentos Formais.	Refinamento e Formalização de Requisitos (Diagramas UML 2.0), <i>Roadmap</i> de Produto.
4	Enquetes, Questionários e Entrevistas.	Refinamento final.

- Considerar as personalidades e estilos de aprendizagem de cada aluno para a atribuição de papéis, tanto na análise de requisitos quanto no projeto de interfaces, leva a uma experiência de aprendizagem mais satisfatória e motivadora para os alunos.

Para determinar a validade das hipóteses apresentadas serão realizadas as seguintes ações:

- Como a matrícula na oficina é opcional aos alunos, serão comparados os resultados obtidos pelos alunos que participaram na oficina com os resultados dos alunos que não participaram. Serão utilizados como parâmetros de comparação: a nota final dos alunos no curso de Metodologia de Análises e uma revisão técnica dos artefatos da etapa de análise, que será realizada pelos alunos que cursam a disciplina de Qualidade de Software.
- Para mensurar a influência dos estilos de aprendizagem e personalidade dos alunos na aplicação das técnicas de DCU, os alunos serão classificados através dos instrumentos definidos em [9,19] e durante a oficina será registrado qual aluno de cada equipe conduz a execução de cada técnica. Finalmente, a qualidade dos protótipos produzidos será avaliada através de avaliação heurística de forma a identificar potenciais relações com os estilos de aprendizagem e personalidade dos alunos.
- Há quatro anos, a universidade vem aplicando um questionário para avaliação da satisfação do aluno sobre cada disciplina, com 27 itens. Os dados obtidos serão utilizados para comparar os resultados da disciplina com a média histórica das disciplinas da área de Gestão de Projetos de Software.

RESULTADOS ESPERADOS

É esperado que os alunos cumpram o objetivo da oficina: identificar, aplicar e avaliar os diferentes métodos utilizados em DCU, mantendo sua relação e contexto com o processo

de desenvolvimento de software. Também se espera atingir os objetivos da disciplina de Metodologia de Análise de forma satisfatória. Também é esperado que a avaliação dos alunos seja positiva, em relação à sua satisfação quanto ao método de ensino, considerando o planejamento diferenciado das atividades a serem realizadas, de acordo com seus estilos de aprendizagem.

CONCLUSÕES E TRABALHOS FUTUROS

Em um futuro imediato, o trabalho se concentra na execução da oficina e na medição contínua dos parâmetros que validarão as propostas dos autores, em relação à distribuição de papéis, estilos de aprendizagem e satisfação dos estudantes com o método de ensino. Conforme mencionado, isso levanta a possibilidade de incorporar revisões dos entregáveis, feitas em paralelo pela disciplina de Qualidade de Software.

Conclui-se que repensar o conteúdo das disciplinas, desde a sua natureza individual até a integração entre elas, pode se constituir como um trabalho potencialmente positivo no processo de ensino-aprendizagem. A revisão de literatura conceitual e prática e os cruzamentos entre as disciplinas também abrem um espaço de experimentação, que permite aprofundá-las e entender melhor como os alunos aprendem.

REFERÊNCIAS

1. Abran, A. and Moore, J.W. *Guide to the software engineering body of knowledge*. IEEE Computer Society, Los Alamitos, Calif., 2004.
2. Ambler, S.W. *Agile modeling: effective practices for eXtreme programming and the unified process*. J. Wiley, New York, 2002.
3. Barcelos, T.S. and Matos, J.P. Quando abordar IHC: o caso da especialização em desenvolvimento de sistemas no IFSP. *Anais Estendidos do X Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais*, (2011).
4. Beck, K. *Extreme programming eXplained: embrace change*. Addison-Wesley, Reading, MA, 2000.
5. Bouzas, P. *El Constructivismo de Vigotsky: Pedagogia Y Aprendizaje Como Fenomeno Social*. Longseller S.A., 2004.
6. Collazos, C.A., Granollers, T., and Ortega, M. Hacia una Integración de Interacción Humano-Computador en las Estructuras Curriculares a Nivel Iberoamericano. *Revista Internacional de Educación en Ingeniería* 3, (2010), 10.
7. Felder, R.M. and Brent, R. Understanding Student Differences. *Journal of Engineering Education* 94, 1 (2005), 57–72.
8. Felder, R.M. and Silverman, L.K. Learning and Teaching Styles in Engineering Education. *Engineering Education* 78, 7 (1988), 674–681.
9. Felder, R.M. and Soloman, B.A. Learning Styles Questionnaire. <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>.
10. Ferreira, J., Noble, J., and Biddle, R. Agile Development Iterations and UI Design. *IEEE* (2007), 50–58.
11. Garrett, J.J. *The Elements of User Experience: User-Centered Design for the Web*. American Institute of Graphic Arts, 2003.
12. Gould, J.D. and Lewis, C. Designing for usability: key principles and what designers think. *Commun. ACM* 28, 3 (1985), 300–311.
13. Leffingwell, D. *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley, Upper Saddle River, NJ, 2011.
14. Luriiia, A.R., Leontiev, A.N., Vygotskii, L.S., and Benitez, M.E. *Psicología y pedagogía*. Akal, Madrid, 2009.
15. Mackay, W.E. and Tatar, D.G. Introduction to the Special Issue on Video as a Research and Design Tool. *ACM SIGCHI Bulletin* 21, 2 (1989), 48–50.
16. McConnell, S. *Professional software development: shorter schedules, higher quality products, more successful projects, enhanced careers*. Addison-Wesley, Boston, 2004.
17. McInerney, P. and Maurer, F. UCD in agile projects. *interactions* 12, 6 (2005), 19–23.
18. Meszaros, G. and Aston, J. Adding Usability Testing to an Agile Project. *IEEE*, 289–294.
19. Myers, I.B. *The Myers-Briggs type indicator*. Consulting Psychologists Press, 1962.
20. Nielsen, J. and Molich, R. Heuristic evaluation of user interfaces. *ACM Press* (1990), 249–256.
21. Patton, J. *Hitting the target*. ACM Press (2002).
22. Piaget, J. *La psicología de la inteligencia*. Crítica, Barcelona, 1999.
23. Sharp, H., Rogers, Y., and Preece, J. *Interaction design: beyond human-computer interaction*. Wiley, Chichester; Hoboken, NJ, 2007.
24. Snyder, C. *Paper prototyping: the fast and easy way to design and refine user interfaces*. Morgan Kaufmann Publishers, San Francisco, CA, 2003.
25. Spencer, D. and Garrett, J.J. *Card sorting: designing usable categories*. Rosenfeld Media, Brooklyn, N.Y., 2009.
26. Spradley, J.P. *Participant observation*. Holt, Rinehart and Winston, New York, 1980.
27. Wilkinson, N.M. *Using CRC cards: an informal approach to object-oriented development*. Cambridge University Press, New York, 1999.